

Supplementary Materials for pFLFE

Anonymous Author

1 Result Supplementations

The domain adaptation capability of pFLFE. We verify the domain adaptation capability on all 3 tasks. In an unseen client, we extract and freeze their encoders trained on other clients, only fine-tuning the decoders.

In Table 1 and Table 2, we observe clear trends across the other two tasks. FedRep outperforms the single global model framework slightly. This can be attributed to the generality and robustness of the global parts generated by FedRep. Compared to other methods, pFLFE demonstrated distinct performance advantages. This indicates that the shared Encoder in pFLFE exhibits better generalization and robustness, making it well-suited for domain adaptation tasks.

Overall, these findings highlight the importance of designing personalized federated learning methods with strong domain adaptation capabilities. By generating shared parts that possess better generalization and robustness, models can effectively adapt to new data domains and achieve superior performance.

2 Experiments setup

Implementation details. We use the Dice coefficient as the evaluation metric. To ensure the reliability of our experiments. It is a set similarity metric commonly used to calculate the similarity between two samples, with a threshold of [0,1]. In medical images, it is often used for image segmentation, with the best segmentation result being 1 and the worst result being 0. The Dice coefficient calculation formula is as follows:

$$Dice = \frac{2 * (pred \cap true)}{pred \cup true} \quad (1)$$

Among them, *pred* is the set of predicted values, while *true* is the set of ground truth values. And the numerator is the intersection between *pred* and *true*. Multiplying by 2 is due to the repeated calculation of common elements between *pred* and *true* in the denominator. The denominator is the union of *pred* and *true*.

We calculate the average Dice coefficient for each client ($Dice_{ACli}$), the average Dice coefficient for all test images ($Dice_{AImg}$), and the variance of Dice across clients ($VDice_{ACli}$) to evaluate the model's performance and client discrepancy. Their calculation method is as follows:

$$Dice_{ACli} = \frac{1}{N} \sum_{i=1}^N \left(\frac{1}{m_i} \sum_{j=1}^{m_i} Dice_{ij} \right) \quad (2)$$

Table 1: Comparison of federated domain generalization results on poly segmentation.

unseen	Client1	Client2	Client3	Client4	Dice _{ACli} ↑
FedAvg	0.6312	0.4261	0.7231	0.4187	0.5226
SCAFFOLD	0.7111	0.5621	0.6933	0.6002	0.6417
FedProx	0.7176	0.5433	0.7225	0.5427	0.6315
Ditto	0.6447	0.4392	0.7125	0.5243	0.5802
FedRep	0.7432	0.6455	0.6972	0.6327	0.6797
ours	0.7672	0.6954	0.8368	0.7477	0.7618

Table 2: Comparison of federated domain generalization results on prostate segmentation.

unseen	Client1	Client2	Client3	Client4	Client5	Client6	Dice _{ACli} ↑
FedAvg	0.8021	0.7895	0.8519	0.6301	0.8209	0.5611	0.7426
SCAFFOLD	0.8214	0.7921	0.8133	0.6501	0.8509	0.5926	0.7534
FedProx	0.8111	0.7881	0.8644	0.6067	0.8048	0.6023	0.7462
Ditto	0.7962	0.7772	0.8791	0.5901	0.8411	0.5097	0.7322
FedRep	0.8457	0.8234	0.8325	0.6992	0.8192	0.6311	0.7752
ours	0.8721	0.8412	0.8848	0.8971	0.8765	0.6598	0.8386

N is the total number of clients. m_i is the total number of data in client i . $Dice_{ij}$ is the Dice result of the i -th client’s j -th data. $Dice_{ACli}$ calculates the average result for each client, and then calculates the average.

$$Dice_{AImg} = \frac{1}{M} \sum_{i=1}^N \sum_{j=1}^{m_i} Dice_{ij} \quad (3)$$

M is the sum of all participating client images. $Dice_{AImg}$ calculates the total result of all client images, and then calculates the average value of each image.

$VDice_{ACli}$ is the variance calculated based on $Dice_{ACli}$ for each client result.

Image augmentations. For RGB images, the input images are resized to 256×256. They undergo color distortion, which includes a random sequence of brightness, contrast, saturation, hue adjustments, and optional grayscale conversion. Finally, random horizontal flips and Gaussian blur are applied to the processed images. For grayscale images, the input images are resized to 384×384. Unlike RGB augmentation, grayscale images do not undergo brightness adjustments. The augmentation includes random horizontal flip, Gaussian blur, and random vertical flip.

Architecture. As for the architecture, to validate the generality of our framework, we employ Encoder-Decoder models such as U-Net, FCN, Res-UNet, and Unet++. The Projector used in personalized contrastive learning is a two-layer MLP. The first layer consists of a linear layer followed by batch normalization and rectified linear units (ReLU). The final MLP contains only a linear layer and ReLU activation, with an output feature dimension of 256.

Baselines. We compare against centralized training, each client’s local training, and a variety of personalized federated learning techniques as well as methods for learning a single global model and their fine-tuned analogues. Centralized training involves collecting data from all clients and training a single model. Each client’s local training trains a model using its own exclusive dataset. Among the personalized methods, we choose FedRep, LG-FedAvg, APFL, and Ditto. Besides, LC-Fed and FedSM provide effective improvements to the FedRep and APFL methods in personalized federated medical image segmentation, respectively. For global FL methods, we choose FedAvg, SCAFFOLD, and FedProx. To obtain fine-tuning results, we first train the global model for the full training period, then each client then fine-tunes all of the model on its local training data for 10 epochs.